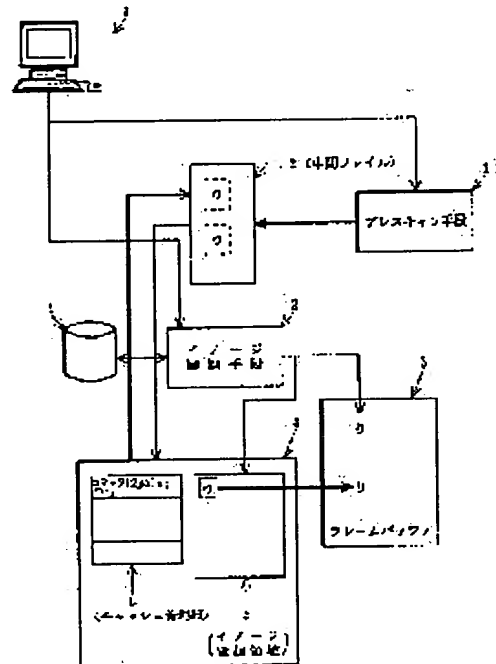


(11)Publication number : 06-186951
(43)Date of publication of application : 08.07.1994

G09G 5/22
G06F 15/20

(72)Inventor : **HARADA MASAHIKO**

CONSTITUTION: Font data for which a computer 7 is to output are prescanned by a prescanning means 11. Consequently, an intermediate file from which the use frequencies, drawing positions, etc., of respective fonts can be found is generated. The data outputted in order from the computer 7 are expanded into image data by an image expanding means 2 and registered in a font cache 4. When the cache 4 becomes full, the intermediate file 12 is referred to and when there is a character which is hit in the intermediate file 12, the whole image data on the hit character are copied to a frame buffer 3. Then the cache 4 is cleared and next expanded data are stored. Thus, similar operation is repeated until all the font data outputted from the computer 7 are copied to the frame buffer 3.



[Date of request for examination]
[Date of sending the examiner's decision of rejection]
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]
[Date of final disposal for application]
[Patent number]
[Date of registration]
[Number of appeal against examiner's decision of rejection]
[Date of requesting appeal against examiner's decision of rejection]
[Date of extinction of right]

7/20/2001 7:0

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-186951

(43)公開日 平成6年(1994)7月8日

(51)Int.Cl. ⁵	識別記号	庁内整理番号	F I	技術表示箇所
G 0 9 G 5/22		8121-5G		
G 0 6 F 15/20	5 6 2 N	9288-5L		
	P	9288-5L		

審査請求 未請求 請求項の数1(全 10 頁)

(21)出願番号 特願平4-353896

(22)出願日 平成4年(1992)12月16日

(71)出願人 000005496

富士ゼロックス株式会社

東京都港区赤坂三丁目3番5号

(72)発明者 原田 政彦

埼玉県岩槻市府内3丁目7番1号 富士ゼ

ロックス株式会社内

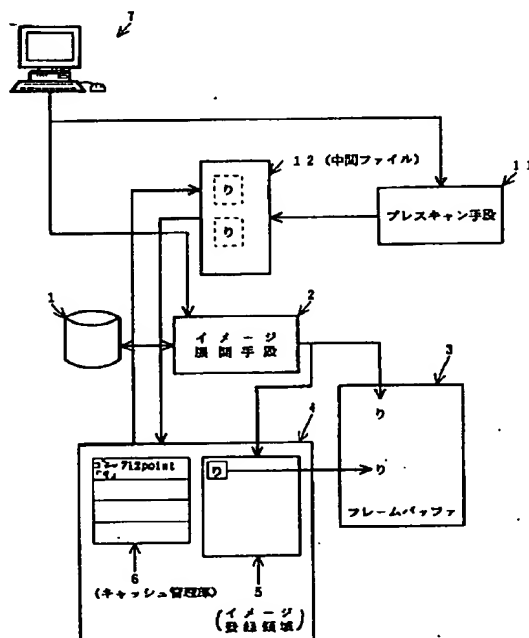
(74)代理人 弁理士 平木 道人 (外1名)

(54)【発明の名称】 文字出力装置

(57)【要約】

【目的】 高速処理ができると共に、フォントキャッシュのメモリ管理を大幅に簡便化した文字出力装置を提供すること。

【構成】 コンピュータ7が出力しようとするフォントデータはプレスキャン手段11でプレスキャンされる。その結果、各フォントの種類と位置がわかる中間ファイル12が作成される。コンピュータ7から順次出力されてきたデータは、イメージ展開手段2でイメージデータに展開され、フォントキャッシュ4に登録される。該キャッシュ4が満杯になると、中間ファイル12が参照され、中間ファイル12中にヒットする文字があると、ヒットした文字全部のイメージデータがフレームバッファ3にコピーされる。その後、キャッシュ4はクリアされ、次の展開データが格納される。以後、コンピュータ7から出力されたフォントデータの全部がフレームバッファ3にコピーされるまで、前記と同様の動作が繰返される。



【特許請求の範囲】

【請求項1】 文字コードデータをイメージデータに変換して出力する文字出力装置において、出力予定のある単位量の文字コードデータをプレスキャンし、各文字コードデータ毎の文字種の情報と位置情報とを検知するプレスキャン手段と、前記プレスキャンされた結果が登録される中間ファイルと、前記文字コードデータが順次展開され登録されるフォントキャッシュと、該フォントキャッシュが満杯になった時に、前記単位量の文字コードデータの中にヒットする文字があるか否かを前記中間ファイルを参照して判断し、ヒットする文字を全て、前記位置情報に基づいて、前記フレームメモリの所定位置にコピーする手段と、該コピー後に、前記フォントキャッシュの内容を削除する手段とを具備し、文字コードデータの展開、フォントキャッシュへの登録、前記単位量の文字コードデータ中のヒットした文字のフレームメモリへのコピー、およびフォントキャッシュの内容の削除を、前記単位量の文字コードデータの書込み又はコピー終了まで繰返すようにしたことを特徴とする文字出力装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 この発明は文字出力装置に関し、特にレーザプリンタ、あるいはパソコン、ワークステーション等のディスプレイに文字を出力するのに使用する文字出力装置に関する。

【0002】

【従来の技術】 従来の文字出力装置の一例を、図5を参照して説明する。図において、1は各文字フォントのベクトルデータをもつフォント辞書、2は文字フォントのベクトルデータを文字のイメージに展開するイメージ展開手段、3は1頁または1画面を単位とする記憶部、例えばフレームバッファ、4はフォントキャッシュである。

【0003】 該フォントキャッシュ4は、展開されたフォントのイメージが記憶されるイメージ登録領域5と、該イメージ登録領域に記憶されたフォントの識別情報を管理するキャッシュ管理部6とから構成されている。7はコンピュータ、8はコンピュータ7からのデータがフォントキャッシュ4に登録されているか否かを判定する登録判定手段である。

【0004】 上記の構成を有する文字出力装置は、図6に示されているような動作をする。すなわち、コンピュータ7からデータを受信すると（ステップS21）、該データが文字コードであるか否かの判断がされる（ステップS22）。文字コードの場合にはステップS23に進む。一方、文字コードでなかった場合にはステップS

24に進んで、対応する処理がなされる。

【0005】 前記ステップS23では、該文字コードがフォントキャッシュ4に登録されているか否かを判定する。この判定は、前記登録判定手段8がキャッシュ管理部6を参照することによって行われる。

【0006】 この判定が肯定の時にはステップS25に進んで、フォントキャッシュ4のイメージ登録領域5内のデータがフレームバッファ3にコピーされる。

【0007】 一方、前記の判定が否定の時には、ステップS26に進んで、イメージ展開手段2を用いて、ビットマップフォントを生成し、フレームバッファ3にコピーする。次いで、ステップS28に進み、フォントキャッシュ4の内容を更新する。

【0008】 ステップS27では、フォントキャッシュ4は満杯か否かの判断がなされ、満杯でない時にはステップS21に戻って、次のデータを受信する。満杯のときには、ステップS29に進み、使用頻度の低い文字が削除される。そして、ステップS30に進み、全データを受信したか否かの判断がなされる。この判断が否定の時には、ステップS21に戻って、次のデータを受信する動作が行われる。一方、ステップS30の判断が肯定になると、前記の処理は終了する。

【0009】 他の従来の文字出力装置として、情報処理学会論文誌、Vol.31 No.1（1990年1月発行）の第144頁～第151頁に開示されたものがある。

【0010】 この先行技術によれば、フォントキャッシュには各種のサイズの文字が登録されるため、単純に文字の登録・削除を繰り返すと、文字のドットイメージを記憶するイメージ登録領域の細分化が発生する。このような状態になると、新しい文字の登録領域を確保するため、イメージ登録領域の未使用領域を1つにまとめるコンパクションを行う必要がある。このコンパクションには、処理時間がかかり、キャッシュの性能低下を招くと開示されている。

【0011】 なお、他の先行技術として、例えば特開平2-150895号公報に開示されたものがある。

【0012】

【発明が解決しようとする課題】 前記した第1の先行技術においては、フォントキャッシュ4が満杯になった時に削除する文字を決定するのに、複雑なアルゴリズムとメモリ管理を必要とするという問題があった。また、フォントキャッシュ4から削除した文字が再度印字データとして出てくる可能性があり、該削除した文字が出てくると、再度ドットイメージに展開し、再びフォントキャッシュ4に登録するといった無駄が生ずるという問題があった。

【0013】 また、第2の先行技術においては、削除した文字のイメージ登録領域の空き領域を有効に使うための再配列（ガベージコレクション）を行うために、複雑なアルゴリズムとメモリ管理を必要とするという問題が

あった。

【0014】本発明の目的は、前記した従来技術の問題点を除去し、高速処理ができると共に、フォントキャッシュのメモリ管理を大幅に簡単化した文字出力装置を提供することにある。

【0015】

【課題を解決するための手段】前記目的を達成するために、本発明は、出力予定のある単位量の文字コードデータをプレスキャンし、各文字コードデータ毎の文字種の情報と位置情報とを検知するプレスキャン手段と、前記プレスキャンされた結果が登録される中間ファイルと、前記文字コードデータが順次展開され登録されるフォントキャッシュと、該フォントキャッシュが満杯になった時に、前記単位量の文字コードデータの中にヒットする文字があるか否かを前記中間ファイルを参照して判断し、ヒットする文字を全て、前記位置情報に基づいて、前記フレームメモリの所定位置にコピーする手段と、該コピー後に、前記フォントキャッシュの内容を削除する手段とを具備した点に特徴がある。

【0016】

【作用】本発明によれば、フォントキャッシュが満杯になった時に、前記単位量の文字コードデータの中にヒットする文字があるか否かが前記中間ファイルを参照して判断される。そして、ヒットした文字は全て、中間ファイルに登録されている前記位置情報に基づいて、フレームメモリの所定位置にコピーされる。

【0017】該コピー後、フォントキャッシュの内容は削除され、次に展開されたデータが登録され、これが満杯になると、前記と同様の動作が繰返される。

【0018】よって、フォントキャッシュの管理を簡単化でき、一度フォントキャッシュに登録された文字が削除された後、再び登録する必要が発生するといった無駄がなくなる。

【0019】

【実施例】以下に、図面を参照して、本発明を詳細に説明する。図1は本発明の一実施例の概略の構成を示すブロック図である。

【0020】図において、1は各文字フォントのベクトルデータをもつフォント辞書、例えばアウトラインフォントデータ辞書、2は文字フォントのベクトルデータを文字のイメージに展開するイメージ展開手段、例えばアウトラインフォント展開手段、3は1頁または1画面を単位とする記憶部、例えばフレームバッファまたは頁バッファ、4はフォントキャッシュである。

【0021】該フォントキャッシュ4は、展開されたフォントのイメージが記憶されるイメージ登録領域5と、該イメージ登録領域に記憶されたフォントの識別情報を管理するキャッシュ管理部（管理テーブル）6とから構成されている。7はコンピュータである。

【0022】また、11はプレスキャン手段、12は中

間ファイルである。該プレスキャン手段11は出力しようとする全データをスキャンして、各文字の文字コード、書体、大きさ等の文字識別情報と、展開されてフレームバッファ上に書かれる文字の位置情報とを検知する。前記中間ファイル12には、前記の文字識別情報と位置情報とが、一時的に記憶される。

【0023】次に、本実施例の動作を、図2のフローチャートを参照して説明する。ステップS1では、コンピュータ7が出力しようとするフォントデータの単位量（例えば、1頁分、または1画面分）がプレスキャン手段11でプレスキャンされ、その結果は、順次中間ファイル12に記憶される。このプレスキャンによって、前記単位量の各フォントの種類と位置がわかる中間ファイル12が作成される。

【0024】次いで、ステップS2に進み、フォントデータがコンピュータ7から受信される。受信後のステップS3～S9の動作は、図6のステップS22～S28の処理と同様であるので、説明を省略する。

【0025】フォントキャッシュが満杯になった後（ステップS8が肯定）、図3のステップS10に進むと、中間ファイル12を参照してフォントキャッシュ4に登録されている文字があるか否か、換言すれば中間ファイル12中に、キャッシュ4にヒットする文字があるか否かの判断をし、ヒットする文字があった場合には、ヒットした文字をフレームバッファ3にコピーする動作が行われる。

【0026】ステップS11では、該コピーが終了したか否かの判断を行い、この判断が否定の時には、ステップS10に戻って、コピーを続行する。ステップS11の判断が肯定になると、ステップS12に進んで、フォントキャッシュの中味を全て削除する。

【0027】ステップS13では、コンピュータ7から出力されたデータのフレームバッファへのコピーが全て終了したか否かの判断がなされる。この判断が否定の時には、ステップS2に戻って、前記した動作が繰返される。

【0028】ステップS13の判断が肯定になると、ステップS14に進んで、プリント出力が実行される。

【0029】以上の動作を、図4を参照して分かりやすく説明する。前記コンピュータ7が出力しようとする1頁分のデータが、21に示されているように、「こんにちはわたしのなまえはたろうです。こんどにちょうびに、」であったとすると、前記ステップS1においてデータ21をプレスキャンすることにより、22に示されているように、どの位置に、どの文字を書くかという情報が、前記中間ファイル12に作成される。

【0030】次に、23に示されているように、前記データ21が1文字ずつ、デコンポーザで展開され、フォントキャッシュ24に登録される。フォントキャッシュ24には「こ」、「ん」、「に」、「ち」の順に、管理

テーブルとイメージ登録領域に登録される。また、これらの文字はフレームバッファに書かれる。

【0031】いま、フォントキャッシュの容量が4文字分しかなく、「こ」、「ん」、「に」、「ち」の4文字の登録により満杯になったとすると、22の中間ファイルが参照され、他に、該4文字と同じ文字、すなわちヒットした文字が前記1頁分のデータ中にないかどうか調べられる。そして、同じ文字があった場合には、頁バッファ25上の該文字が書かれる位置にコピーされる。

【0032】図4の例では、「こんにちようびに、」の「こん」、「にち」および「に」が、図示されているようにコピーされる。

【0033】次に、前記フォントキャッシュ24の内容が全て削除され、次のデータ21がデコンポーザで展開され、フォントキャッシュに登録される。すなわち、フォントキャッシュ24には、次の4文字「は」、「わ」、「た」、「し」が、管理テーブルとイメージ登録領域に登録されかつフレームバッファに書かれる。

【0034】次いで、中間ファイルが参照され他に同じ文字があった場合には、フレームバッファ上の該当位置にコピーされる。

【0035】以上の動作が繰り返され、フレームバッファに全部のデータのイメージデータが書込まれる又はコピーされると、プリント動作が実行される。

【0036】以上のように、本実施例によれば、フォントキャッシュが満杯になった時に、プリントをする予定のデータ中にヒットする文字があるか否かを調べ、ヒットする文字は全てフレームバッファにコピーし、その後フォントキャッシュをクリアするようにしたので、フォントキャッシュの管理を簡単化することができる。また、一度フォントキャッシュに登録された文字が削除された後、再び登録するといった無駄がなくなる。

【0037】なお、前記の実施例はレーザプリンタばかりでなく、パソコン、ワークステーション等のディスプレイにも応用でき、高速な文字出力を期待することができる

05 【0038】。

【発明の効果】本発明によれば、フォントキャッシュが満杯になった時に、プリントをする予定のデータ中にヒットする文字があるか否かを調べ、ヒットする文字は全てフレームバッファにコピーし、次いでフォントキャッシュの内容を全部削除し、新たに次のデータを登録するようにしたので、フォントキャッシュの管理を簡単化できるという効果がある。

10 【0039】また、一度フォントキャッシュに登録された文字が削除された後、再び登録するといった無駄がなくなり、高速な文字出力を期待することができる。

15 【図面の簡単な説明】

【図1】 本発明の一実施例の概要を示すブロック図である。

20 【図2】 本実施例の動作を説明するフローチャートである。

【図3】 図2の続きのフローチャートである。

【図4】 本実施例の一具体例の説明図である。

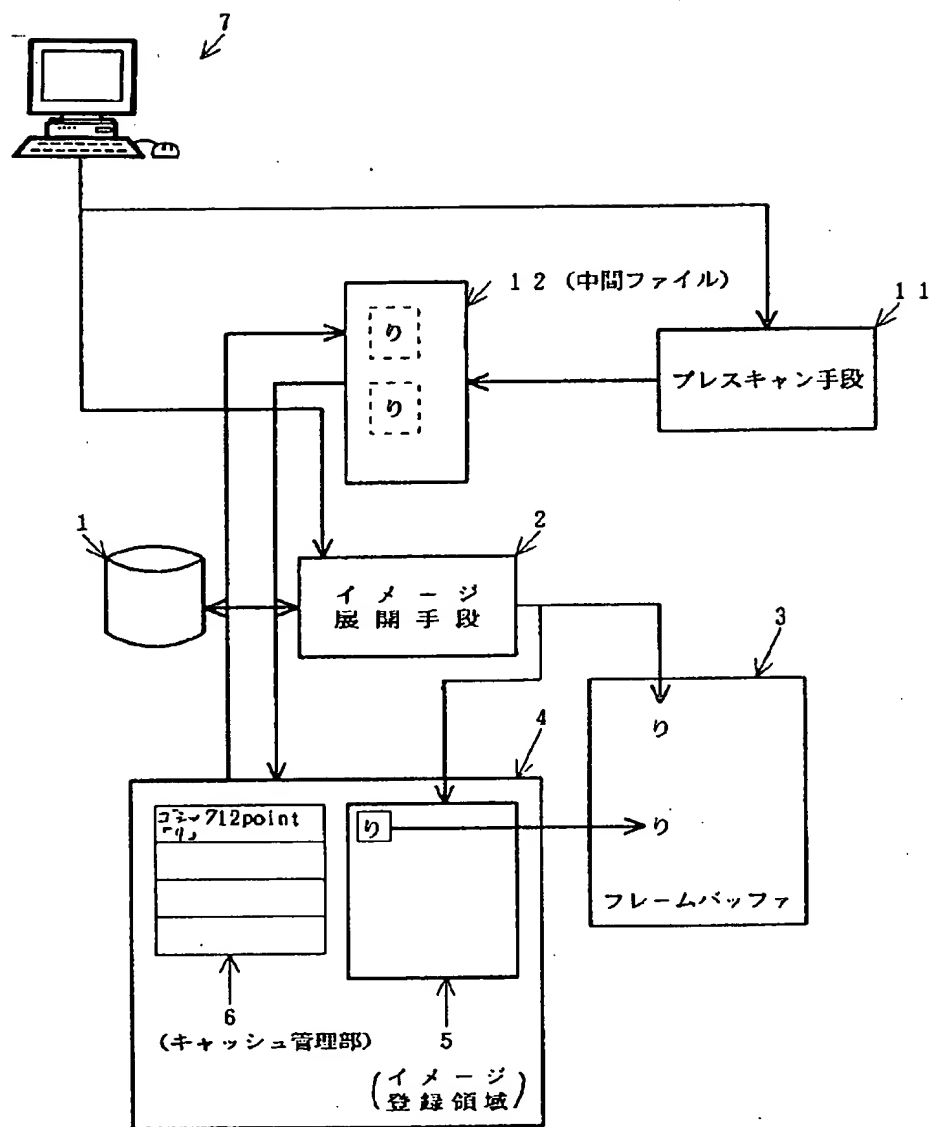
25 【図5】 従来装置の構成の概要を示すブロック図である。

【図6】 従来装置の動作を説明するフローチャートである。

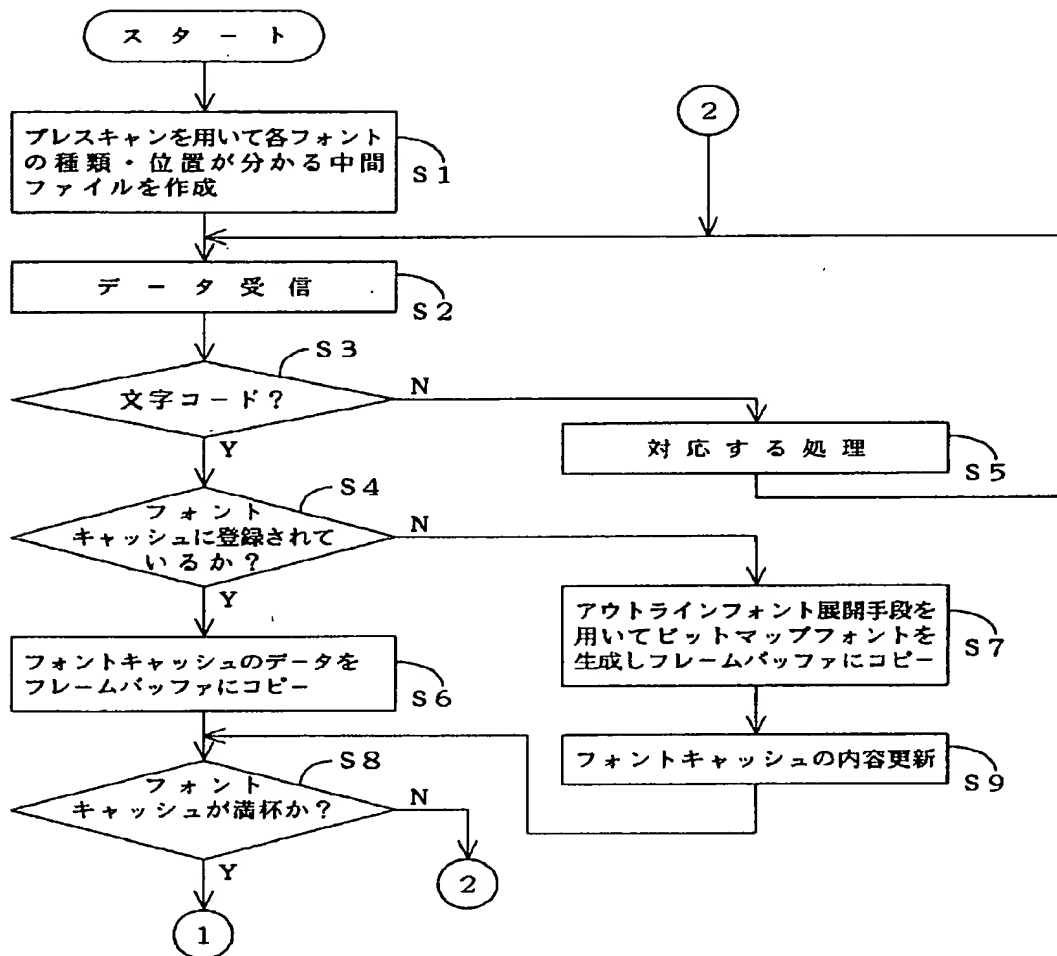
【符号の説明】

30 1…フォント辞書、2…イメージ展開手段、3…フレームバッファ、4…フォントキャッシュ、5…イメージ登録領域、6…キャッシュ管理部（管理テーブル）、7…コンピュータ、11…プレスキャン手段、12…中間ファイル。

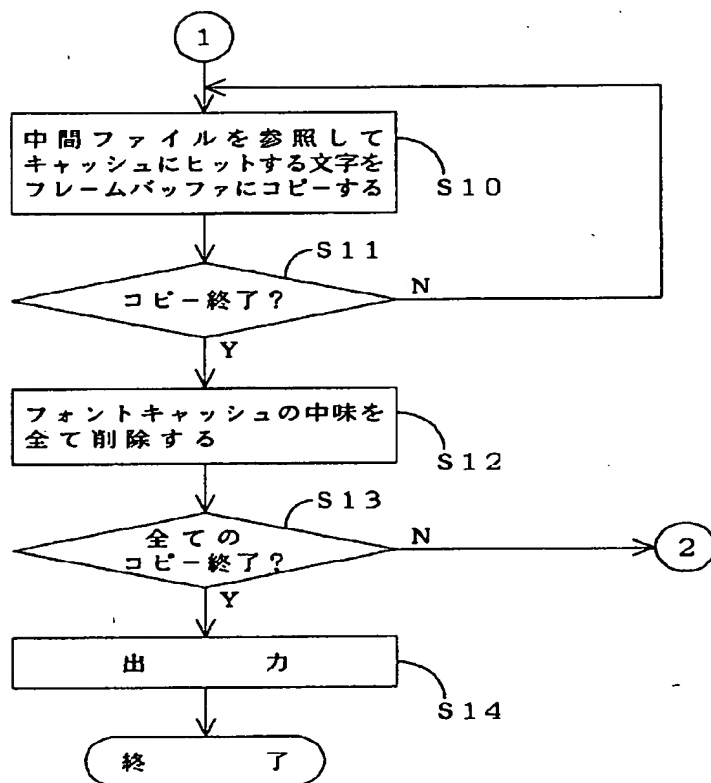
【図1】



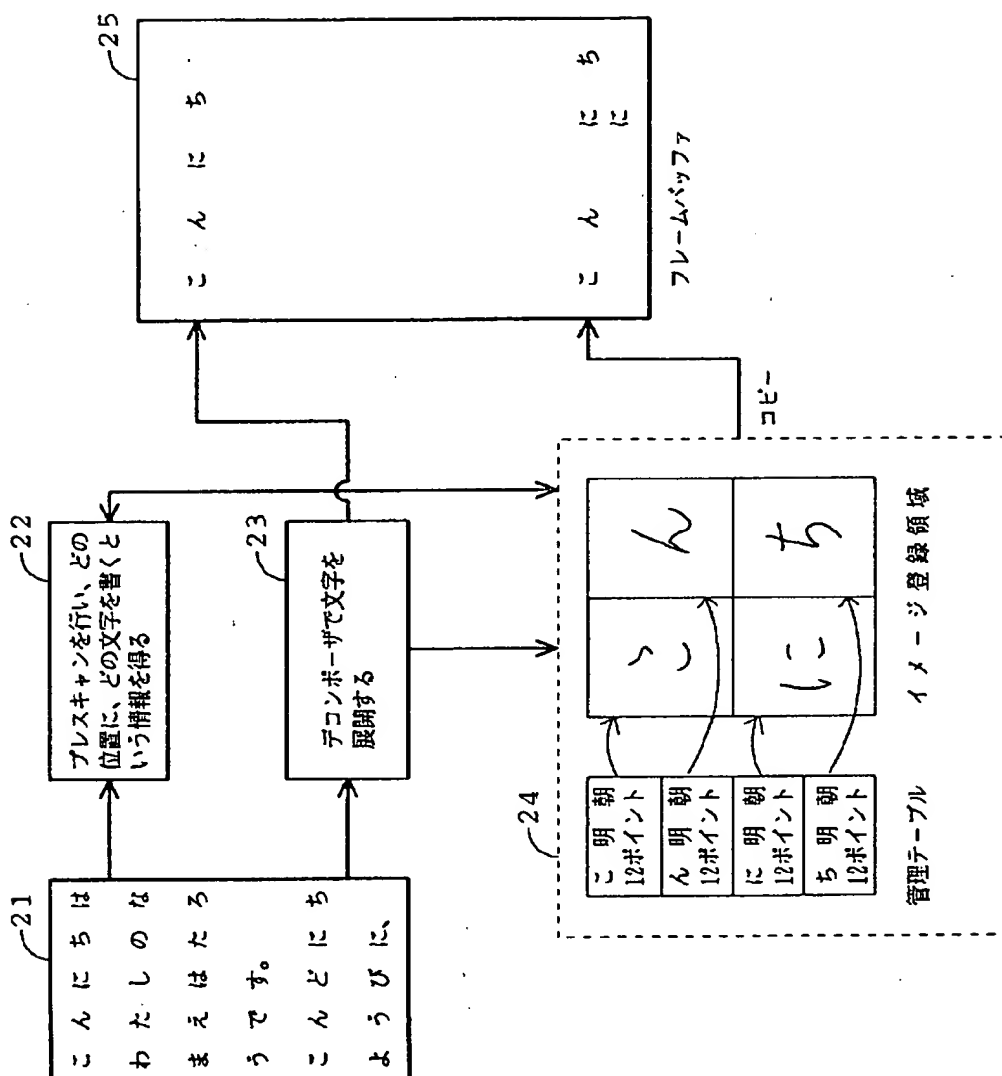
【図2】



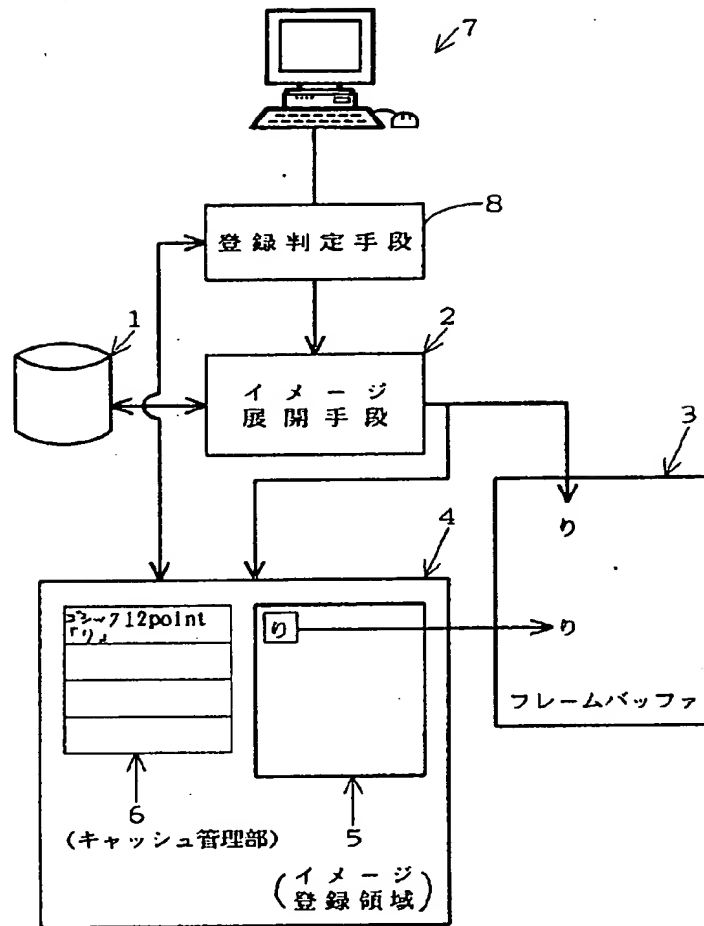
【図3】



【図4】



【図5】



【図6】

